Installing Applications in FreeBSD

Reference: NYCU CSCC SA Course

國立成功大學資訊工程系

Department of Computer Science and Information Engineering, NCKU

Handbook and Manual pages

- Complete guide and be found at
  - https://www.freebsd.org/doc/handbook/ports.html
  - https://www.freebsd.org/doc/zh_TW/books/handbook/ports.html
  - ports(7)
  - pkg(7), pkg(8)

Before we start

- Permission issue
  - root: the superuser
    - *In Unix-like system, root is the conventional name of the user who has all rights or permissions (to all files and programs) in all modes (single- or multi-user)*
- Don't execute any command as root directly
  - It's DANGEROUS
- However sometimes you still need to be root to do something
  - Install software
  - Manage system settings
  - Create/modify/delete users

Before we start

- Become root
  - Console login with root
  - By default, you cannot login as root via SSH
- Change current user
  - Don't need to login with console
  - Use command "su -", and then type root's password
    - Only user in "wheel" group can use "su -"
  - To see which account you are using, use "whoami"

```
$ whoami
tsaimh
$ su -
Password:
$ whoami
root
```

Before we start

- As mentioned before, don't run as root directly
- Can we execute with root's credential only for some specific commands?
    - Like 'Run as administrator' in Windows
    - Is there similar commands in Unix-like system/FreeBSD?

Before we start

- Run commands with other user's permission
- "sudo" command
  - Only simplest explanation here for basic usage
  - "sudo" syntax and other details will be explained in later chapters
  - Here only tell you how to simply enable 'sudo'
- How to enable sudo?
  - "sudo" is not a command in the base, needs to be installed manually

Before we start – Enable "sudo" (1)

- Install the package
  - Check Internet connection
    - `$ ping -c4 8.8.8.8`
  - Become root
    - `$ su -`
  - Install the package of sudo
    - `$ pkg install sudo`
      - This will install 'sudo' from Internet
      - Type ‘Y’ (means yes) when it asks for confirmation

Before we start – Enable "sudo" (2)

- Allowing your user to execute "sudo"
  - Switch to root first
  - If you are not familiar with the default editor 'vi', type the following command to change your editor for this time (skip this step otherwise)

    ```
    $ setenv EDITOR ee
    ```

    - Will explain this in later chapters
    - This will allow you using a notepad-like editor

Before we start – Enable "sudo" (3)

- Allowing your user to execute "sudo"
    - Type "visudo" to edit the sudoer file
        - Specify who can use "sudo"

```
##
## User privilege specification
##
root ALL=(ALL) ALL
tsaimh ALL=(ALL) ALL
```

- Save the file and exit, back to normal user
    - Use "logout" command or press Ctrl+D

Before we start – Using "sudo"

- Now, you can prepend "sudo" before commands to run them as root
    - But please think carefully before you hit enter
- Execute commands with "sudo"
    - sudo whoami
        - You have root's credential
    - sudo pkg install vim
        - Install software without becoming root directly
    - You need to re-type your password
        - Don't need to re-type within 5 minutes

Install software: Overview (1)

- Package (Pre-built binary programs)
  - Like installers (.msi) in Windows
  - "package" (.txz) on FreeBSD
  - rpm on RedHat Linux, deb on Debian Linux
- Package Manager
  - install/remove/upgrade packages
  - Other Unix-like systems
    - rpm, yum, dpkg, apt, dnf, pacman …
  - FreeBSD
    - pkg

Install software: Overview (2)

- Install from source
  - Managed source collection
    - FreeBSD Ports
    - With dependency checking and FreeBSD specified patches
  - Others
    - Download source tarball (.tar.gz) from websites
    - Checkout from VCS (git/svn)
    - No dependency checking

Install software: Comparison (1)

Method	Description	Dependency Checking
Packages	Pre-built ports, contains pre-compiled copies of all the commands for the software with default settings, as well as any configuration files or documentation.	Yes
Ports	A collection of files designed to automate the process of compiling an software from source code and additional patches (a set of Makefile, patches, description files, …)	Yes
Tarball VCS	fetch it, configure the installation options, and compile it by yourself.	No

Install software: Comparison (2)

Method	Benefits
Packages	● Packages do not require any additional compilation ● Benefit for slow machines
Ports	● Optimization ○ You can tweak the compilation options to generate code that is specific to a different processor ● Customization ○ Some software have compilation time options relating to what they can and cannot do
Tarball VCS	● Some software cannot be found in ports collection ○ Newly created projects, latest versions, ... ● Some latest versions of software may have new configurations that do not exist in ports (cannot configure it through the ports easily)

Package System (1)

- pkg
  - New generation of FreeBSD package system
- Install new software
  - Fetch packages from a repository
  - Need root permission (sudo)
  - Automatically update the database
    - By default invoking either of pkg install or pkg upgrade will cause repository catalogues to be updated automatically
  - Perform dependency check
    - Will install software that required by new software

Package System (2)

- Install new software
  - pkg install <names of packages…>
    - pkg install vim-console tmux
- Upgrade currently installed software
  - pkg upgrade <names of packages…>
    - pkg upgrade vim-console
  - pkg upgrade
    - Upgrade all installed software
  - This will also update the database

Package System (3)

- Update packages database only
  - pkg update
- Delete a package
  - pkg delete <names of packages>
- Search
  - pkg search <keyword>
  - Search package repository catalogues

Package System (4)

- Show information about installed packages
  - pkg info
    - Show all installed packages
    - Use "grep" to find specific packages
      - pkg info | grep vim
  - pkg info <name of package>
    - Show detailed information
    - pkg info vim-console

Package System (5)

- Show version of installed packages
  - pkg version
    - pkg version -v

```
$ pkg version -v
bash-4.3.46_1              < needs updating (remote has 4.4.12_2)
bind99-9.9.9P8_1           < needs updating (remote has 9.9.10P3)
ca_root_nss-3.32           = up-to-date with remote
```

Port System

- We should…
    - Obtain the ports collection
        - List of ports available to be installed into system
    - Find the application
    - Change to the directory for the port
- Ports will
    - Fetch the source tarball
    - Ask for configuration friendly
    - Compile the source code to a **package**
    - Install the application via the just built package
- Deinstall process

Obtaining the Ports Collection (1/3)

- portsnap(8)
  - Fetch and update your ports tree
  - fetch, extract, update, cron
  - sudo portsnap fetch extract update
  - https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ports-using.html

Obtaining the Ports Collection (2/3)

- git (1)
  - Install git command line tool
    - sudo pkg install git
  - Checkout from a given repository
    - sudo git clone https://git.FreeBSD.org/ports.git /usr/ports

Obtaining the Ports Collection (3/3)

- Port directory
  - /usr/ports/<category>/<name>

```
$ ls /usr/ports/
CHANGES         archivers       finance         multimedia      textproc
CONTRIBUTING.md astro           french          net             ukrainian
COPYRIGHT       audio           ftp             net-im          vietnamese
GIDs            base            games           net-mgmt        www
Keywords        benchmarks      german          net-p2p         x11
LEGAL           biology         graphics        news            x11-clocks
…

$ ls /usr/ports/editors/vim
Makefile        distinfo        files           pkg-descr       pkg-plist
```

Ports system (1)

- Find your application
  - cd /usr/ports
  - For the first time, run "sudo make fetchindex" to fetch index for searching
  - make search name=program name
  - make search key=string

```
$ make search name=vim-console
Port:   vim-console-8.2.1558
Path:   /usr/ports/editors/vim-console
Info:   Improved version of the vi editor (console only)
Maint:  adamw@FreeBSD.org
B-deps: pkgconf-1.7.3,1
R-deps:
WWW:    http://www.vim.org/
```

Ports system (2)

- psearch(1)
  - Simple but useful tool to find ports
  - ports-mgmt/psearch
    - or pkg install psearch
  - psearch <name of port>
    - psearch vim

```
$ psearch vim
audio/vitunes                Curses-based media player with vim-like keybinds
devel/clewn                  Clewn provides Gdb support within Vim
editors/cream                Gvim extension with many features
...
```

Ports system (3)

- Type "make install clean" to install your application
  - make config (/var/db/ports/)
  - make fetch (/usr/ports/distfiles/)
  - make checksum
  - make extract
  - make patch
  - make configure
  - make build

    make (all)
  - make install
  - make clean
    - Clean files generated by configure process
  - make distclean
    - Clean downloaded distribution files (tarball)

Ports system (4)

- The ports system uses fetch(1) to download the files
  - MASTER_SITES environment variable
  - /etc/make.conf

```
MASTER_SITE_BACKUP?=    \
       http://FreeBSD.cs.nctu.edu.tw/distfiles/${DIST_SUBDIR}/
MASTER_SITE_OVERRIDE?=  ${MASTER_SITE_BACKUP}
```

- Options for ports
  - make config
    - Won't build or install the port
    - Use this to re-configure ports (otherwise, it uses old one instead)
  - hidden options (not shown in 'make config')
    - Edit the Makefiles under that port directory

Ports system (5)

- I have installed the application but Command not found…
  - Logout, and then login.
  - If you use (t)csh or zsh
    - rehash

Upgrading Ports using Portmaster

- ports-mgmt/portmaster
  - A utility for easily upgrading and installing ports

```
$ cd /usr/ports/ports-mgmt/portmaster && make install clean
```

- Install or upgrade a port
  - portmaster <category>/<name>
    - portmaster sysutils/lsof
  - /usr/ports/UPDATING
    - Read before attempting any port upgrades!!!
- Useful options
  - -B, -D, -a, -r, -y, -H, -w
  - portmaster -dyBwH editors/vim
  - /usr/local/etc/portmaster.rc

Security

- Show security issues about installed packages
  - No matter from port or from package
  - pkg audit
  - Upgrade these packages to mitigate security problems

```
$ pkg audit
python38-3.8.10 is vulnerable:
  Python -- multiple vulnerabilities
  WWW: https://vuxml.FreeBSD.org/freebsd/145ce848-1165-11ec-
ac7e-08002789875b.html
```

Install from source (1)

- Compile the source files first and then install
  - Tarball, a pack of source code
    - tar -xzf certain-source.tar.gz
    - cd certain-source
    - ./configure [options …]
      - ./configure --help
    - make
    - make install (root permission)

Install from source (2)

- Compile the source files first and then install
  - Checkout master branch from VCS
    - git clone --depth=1 https://github.com/curl/curl.git
    - cd curl
    - ./buildconf
    - ./configure --enable-debug
    - make
    - sudo make install

Security Considerations (1)

- How to find secure source
  - Check the official site, read the announcement and change log
  - Verify the checksum (tarball)
  - Fetch via https or ssh (VCS)

Security Considerations (2)

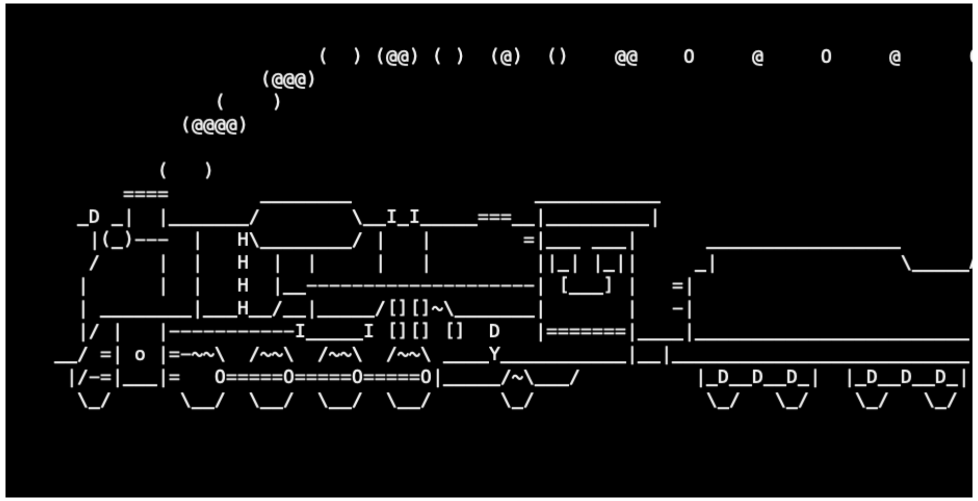
- Why "curl *URL* | sh" is bad?
  - Example: "curl get.pow.cx | sh"
  - Why do you think this is good?
  - Search: "curl pipe bash"
    - Even the file does not contain evil code, broken connection may turn it.
    - "rm -rf /tmp/foo.bar" becomes "rm -rf /"
  - Instead: download the script, read it, execute it.

Deinstall Applications

- Two methods
  - pkg delete
    - Find the package name via pkg info
    - Dependency check
    - Disable dependency check
      - -f : force
      - pkg delete -f <names of packages>
  - make deinstall
    - Change to the port’s directory
    - make deinstall
    - Delete it anyway
    - Similar to "pkg delete -f"

Try to install from ports/pkg

- tmux
- vim-console, emacs
- mutt
- wget, curl
- lftp
- lynx, w3m
- expect
- zsh, bash
- ~~sl~~



Appendix

Package management in other Unix-like systems

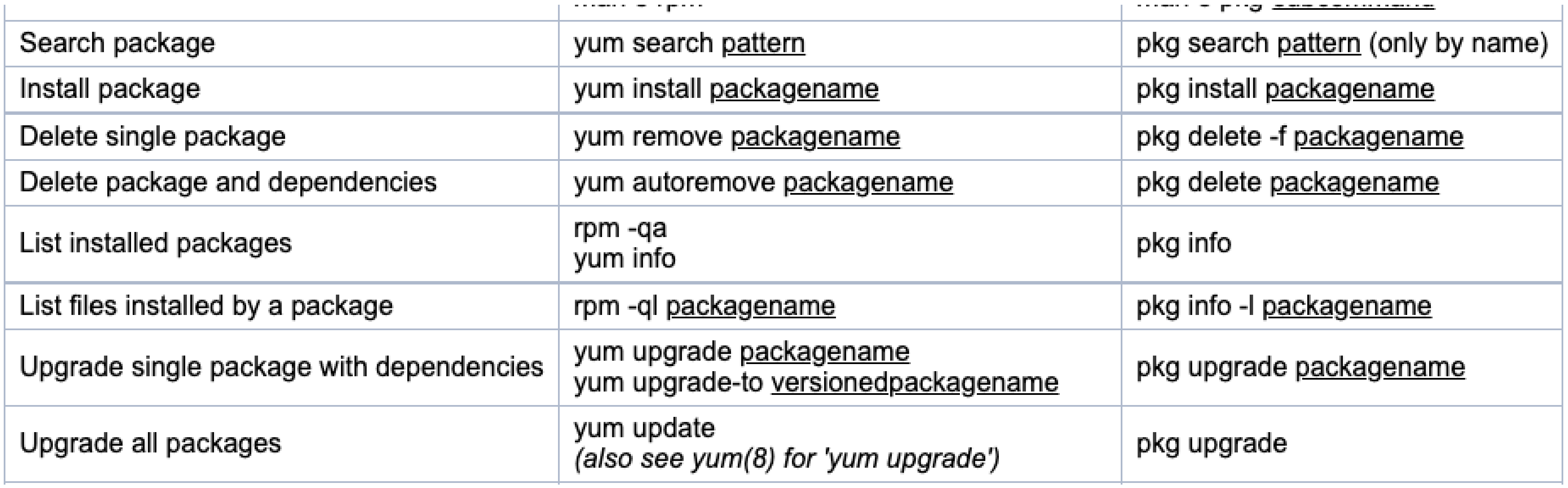
Reference: NYCU CSCC SA Course

國立成功大學資訊工程系

Department of Computer Science and Information Engineering, NCKU

Package Manager Rosetta Stone

- Package commands in the most common systems
- https://wiki.freebsd.org/PackageManagerRosettaStone



	man 8 rpm	man 8 pkg subcommand
Search package	yum search pattern	pkg search pattern (only by name)
Install package	yum install packagename	pkg install packagename
Delete single package	yum remove packagename	pkg delete -f packagename
Delete package and dependencies	yum autoremove packagename	pkg delete packagename
List installed packages	rpm -qa yum info	pkg info
List files installed by a package	rpm -ql packagename	pkg info -l packagename
Upgrade single package with dependencies	yum upgrade packagename yum upgrade-to versionedpackagename	pkg upgrade packagename
Upgrade all packages	yum update *(also see yum(8) for 'yum upgrade')*	pkg upgrade